Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Estatística

Estudo comparativo de heurísticas para resolução de problemas MSSC utilizando o Sistema de Operação R

Felipe Veloso Alves Carreiro
e Lucas César Pereira Vale

Trabalho de Conclusão de Curso apresentado ao Departamento de Estatística da Universidade de Brasília, como parte dos requisitos para a obtenção do título de Bacharel em Estatística.

Brasília
2016

FELIPE VELOSO ALVES CARREIRO
e LUCAS CÉSAR PEREIRA VALE

Estudo comparativo de heurísticas para resolução de problemas MSSC utilizando o Sistema de Operação R

Trabalho de Conclusão de Curso apresentado ao Departamento de Estatística da Universidade de Brasília, como parte dos requisitos para a obtenção do título de Bacharel em Estatística.

Orientador: Profa. Dra. **Maria Amélia Biagio**

Brasília
2016

Dedicatória

Dedicamos este trabalho primeiramente à Deus.
Dedicamos também aos nossos pais, irmãos,
familiares e amigos.

Felipe Carreiro e Lucas Vale

Agradecimentos - Felipe Carreiro

Agradeço primeiramente a Deus por ter me ajudado a fazer este trabalho, e me dado força para superar todas as dificuldades.

Á minha orientadora Maria Amélia pela paciência, apoio e confiança na elaboração do trabalho.

Aos meus pais Firmino e Eliene, pela educação, amor e incentivo que sempre me dedicaram.

Aos meus irmãos, Danilo e Lucas, por toda ajuda e disposição que tiveram comigo.

Aos meus demais familiares, minha avó, meus tios e primos, por toda preocupação e apoio.

Aos professores e funcionários da Universidade de Brasília que contribuíram de alguma forma para minha formação acadêmica.

Aos meus amigos e colegas que conheci durante o curso, são muitos e não tem como citar todos, mas em especial aos meus amigos Erique, por toda amizade desde o primeiro dia de aula, sempre estando presente ao longo da minha graduação e me incentivando, ao Felipe Quintino por toda ajuda no trabalho, disponibilizando seu tempo e sempre mostrando solidariedade quando estava precisando, não somente no trabalho realizado, mas durante todo o curso e ao Lucas César por todo o companheirismo, paciência e alegria que tivemos ao longo do trabalho e esses anos.

A todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

Agradecimentos - Lucas Vale

A Deus por ter me dado saúde e força para superar as dificuldades.

Aos meus amigos que caminharam junto comigo desde o início da faculdade e que sempre me deram apoio. Ao Felipe Veloso, meu parça que sempre esteve presente em toda a caminhada, e ao Felipe Quintino por toda dedicação e esforço para nos ajudar no desenvolvimento do trabalho.

À minha orientadora Maria Amélia, por todo o suporte, pelas suas correções e incentivos.

Aos meus pais, pelo amor, incentivo e apoio incondicional.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

"Todo tempo é tempo de acreditar que as pessoas vão se renovar."

São João Paulo II

Resumo

Metodologias não-hierárquicas para agrupamento de dados têm sido bastante estudadas e utilizadas nas últimas décadas, muitas delas buscando otimizar um critério comum que é o de minimizar a soma dos quadrados das discrepâncias internas aos grupos formados por seus procedimentos. O problema de se agrupar dados com este critério é bastante conhecido na literatura como problema *MSSC* (*Minimum Sum of Squares Clustering*). Dentre as metodologias voltadas para a resolução de problema *MSSC* deve-se citar a já bastante conhecida heurística *K-Means*. Com o mesmo propósito, muitas metodologias surgiram nas últimas décadas e, dentre as principais, destacam-se as metodologias *H-Means*, e mais recentemente sua forma não-degenerada *H-Means+*, as metodologias *Tabu Search* e *VNS*. No entanto, em ambiente computacional fortemente demandado por estatísticos, como o sistema computacional R, estas metodologias, com exceção da primeira, ainda não estão disponíveis. O presente trabalho consiste no estudo dessas novas heurísticas e na modificação da versão implementada da Busca Tabu e *H-Means* para melhorar os resultados já conhecidos com alguns bancos de dados. Resultados computacionais são obtidos para os bancos de dados *USArrests* e Íris de Fisher, ambos disponíveis no mesmo sistema em referência. Análise comparativa dos agrupamentos, obtidos pelas metodologias *K-Means*, a versão implementada e modificada da *HBaseTabu* e do *H-Means* apresentada para distintos números de *clusters*. Os resultados apresentados são validados, no primeiro teste, através dos valores ótimos apresentados por *K-Means*, e por valores ótimos já conhecidos para os testes realizados com o banco de dados Íris. Através da análise dos resultados obtidos, pode-se observar que a heurística implementada neste trabalho apresenta resultados melhores daqueles obtidos por *K-Means* e demonstram, em vários casos, superioridade sobre as demais heurísticas, o que mostra o poder de eficiência das mudanças realizadas na implementação computacional desse novo algoritmo.

Palavras-chave: Agrupamento de Dados, problemas *MSSC*, heurística Busca Tabu, linguagem de computação R, *VNS*.

Sumário

Lista de Figuras

Lista de Tabelas

Introdução

O problema de agrupamento de dados é bastante conhecido nas áreas da matemática, estatística e áreas afins. Como exemplos, pode-se citar disciplinas como biologia, botânica, medicina, psicologia, geografia, *marketing* e processamento de imagem. Vários estudiosos propuseram metodologias para a resolução do problema de agrupamento de dados. Dentre eles deve-se citar alguns pioneiros no assunto como Johnson (2007), Hohlf (1973,1978), Lance e Williams (1967), e Ward (1963) (vide Kaufman e Rousseeuw, 1990). Estas metodologias se dividem em dois grandes grupos: as Hierárquicas e as Não-hierárquicas.

As metodologias Hierárquicas são classificadas em Aglomerativas e Divisivas. Já bastante conhecida na literatura, a mais utilizada é a Aglomerativa por sua maior simplicidade e eficiência computacional, em que se procede uma série de sucessivas fusões dos n indivíduos em grupos. Por outro lado, os métodos Divisivos separam os n indivíduos sucessivamente em agrupamentos mais finos.

As metodologias Não-hierárquicas, por apresentarem maior flexibilidade computacional que as Hierárquicas, vem sendo exaustivamente estudadas com o objetivo de se alcançar melhores soluções para o problema de Agrupamento de Dados. Este problema é comumente formulado e resolvido como um problema *MSSC (Minimum Sum of Squares Clustering)*, que possui característica altamente combinatorial. Dentre as metodologias mais conhecidas, e que apresentam solução para o problema *MSSC* destacam-se as heurísticas *K-Means* (McQueen, 1967) e *H-Means* (Forgy, 1965). A resolução deste problema não é trivial, dado que o mesmo possui muitos mínimos locais, e este fato tem estimulado estudiosos do assunto a proporem metodologias que sejam capazes de apresentar soluções próximas da solução ótima global. Para tanto, as heurísticas Busca Tabu (ou *Tabu Search*) e *VNS* (*Variable Neighborhood Search*) têm sido amplamente utilizadas em problemas de distintas áreas do saber.

Com o propósito de estudar estas heurísticas em ambiente do Sistema Computacional Estatístico R, o presente trabalho distribui-se da seguinte forma: no capítulo 1 apresenta Aspectos do Estado da Arte, no capítulo 2 as metodologias estudadas e re-

formuladas, no capítulo 3 os resultados obtidos e análise comparativa entre os métodos em estudo e no capítulo 4 conclusões acerca dos resultados obtidos e as considerações finais do trabalho.

Objetivo Geral

O objetivo do trabalho é buscar formulação adequada para uma heurística que utiliza princípios da metodologia Busca Tabu (ou *Tabu Search* - TS) na busca de soluções globais para o problema *MSSC (Minimum Sum of Squares Clustering)*, e explore a heurística *H-Means* para alcançar soluções locais.

Objetivos específicos

Um estudo comparativo desta nova metodologia com a metodologia *VNS*, recentemente implementada em linguagem R (Quintino, F., julho 2015) e outras metodologias clássicas de agrupamento de dados que estão disponíveis no mesmo sistema, foram estudadas e serão apresentadas. Para isso deve ser elaborado um programa computacional em linguagem compatível àquela do sistema de operação estatística R. Além do aprendizado sobre algumas das principais metodologias para agrupamento de dados, espera-se obter boa adaptação da *Tabu Search* ao ambiente R e que as soluções indicadas por esta versão sejam tão boas ou melhores que aquelas indicadas por outros métodos disponíveis e/ou implementados em linguagem do sistema R.

Capítulo 1

Aspectos do Estado da Arte

Dada a natureza combinatorial do problema de agrupamento de dados, o *boom* da tecnologia da informática estimulou enormemente o desenvolvimento de estudos e pesquisas relacionados à resolução deste problema. Consequentemente, nas últimas décadas, muitos foram os estudiosos que propuseram metodologias para a resolução do problema em referência; dentre eles deve-se citar alguns pioneiros no assunto como Florek et al. (1951), Johnson (1967), Lance e Williams (1967), e Ward (1963) (vide Kaufman e Rousseeuw,1990). Levantamentos bibliográficos sobre as várias metodologias existentes podem ser encontrados em Aloise e Hansen, (2008) e Jain, (2010).

Todas as metodologias para agrupamento de dados necessitam utilizar medidas de similaridade (ou dissimilaridade) entre os elementos estudados e um critério para obtenção dos grupos, os quais devem ser escolhidos de acordo com a natureza (se qualitativos e/ou quantitativos) e disposição dos elementos no conjunto de dados. Elas diferem bastante entre si e podem ser divididas, segundo a literatura, em dois grandes grupos: as Hierárquicas e as Não-hierárquicas.

As metodologias Hierárquicas necessitam a princípio da informação sobre a matriz de similaridade (ou dissimilaridade) entre os elementos observados e pertencentes ao conjunto de dados. Elas são classificadas em Aglomerativas e Divisivas. As Aglomerativas tomam, inicialmente, tantos grupos quantos forem os elementos a serem estudados e, a cada passo, de acordo com um critério, une aqueles mais similares para formarem um novo grupo, e assim segue o processo até obter apenas um grupo com todos os indivíduos do conjunto de dados. De acordo com o critério de aglomeração utilizado, as metodologias Hierárquicas Aglomerativas recebem as seguintes denominações: Ligação do Vizinho mais Próximo, Ligação do Vizinho mais Longe, Média dos Grupos e a de Ward.

Com procedimento inverso ao utilizado pelas últimas, as Hierárquicas Divisivas inicializam o procedimento com um grupo formado por todos os elementos em estudo e, passo a passo, procede na divisão do grupo mais dissimilar em outros dois

grupos a fim de obter, no final, tantos grupos quanto forem os elementos do conjunto de dados. Este processo de divisão dos grupos demanda grande carga computacional; por esta razão, as metodologias Aglomerativas tem sido preferidas por apresentarem maior simplicidade e eficiência de implementação.

As metodologias Não-hierárquicas não necessitam da informação da matriz de similaridade para iniciarem o procedimento mas requerem, como parâmetro, o número de agrupamentos, e buscam encontrar a melhor partição, para aquele número, existente no conjunto de dados. Nestas, iterativamente, as partições são obtidas de maneira a otimizar um determinado critério, que possui como objetivo alcançar maior homogeneidade dentro grupos (ou maior heterogeneidade entre os grupos). Dessa forma, passo a passo, elementos pertencentes a um grupo podem ser designados a outros grupos desde que o critério supracitado seja satisfeito. Assim, por apresentarem menor exigibilidade computacional que as Hierárquicas, as metodologias Não-Hierárquicas vem sendo exaustivamente estudadas com o objetivo de se tentar alcançar melhores soluções para o problema de agrupamento de dados.

Um dos objetivos comumente utilizado pelas metodologias Não-Hierárquicas é a minimização da soma dos quadrados das discrepâncias dentro dos grupos; com isso, o problema de encontrar a melhor partição em um conjunto de dados passa a ser denominado, na literatura, como um problema *MSSC (Minimum Sum of Squares Clustering)*. Sua resolução não é trivial, dado que o mesmo possui muitos mínimos locais, e este fato tem estimulado estudiosos do assunto a proporem metodologias que, associadas ou não às Não-hierárquicas, sejam capazes de apresentar soluções próximas da solução ótima global.

Dentre as metodologias mais conhecidas, voltadas para a resolução de problemas *MSSC*, estão as heurísticas *K-Means* (McQueen, 1967) e *H-Means* (Forgy, 1965). O fato deste problema ser de resolução complexa, apresentando muitos mínimos locais, estimulou o surgimento de muitas outras metodologias cujo propósito é o de alcançar soluções melhores, procurando explorar vizinhanças distintas daquela onde um mínimo local é geralmente encontrado. Dentre elas, pode-se citar as heurísticas *VNS* (Hansen e Mladenovic, 2001) e Busca Tabu (Al Sultan, 1995, Glover e Laguna, 2002), que possuem como ponto comum a exploração de vizinhanças cada vez mais distantes daquela onde se encontra um mínimo local conhecido. Para alcançar regiões distintas do conjunto solução do problema em questão, estas últimas constroem novas partições do conjunto de dados a partir de perturbações da partição correspondente a uma determinada solução, a qual, para a Busca Tabu, pode ou não ser um mínimo local.

A metodologia *Variable Neighborhood Search (VNS)* é uma heurística recente, proposta por Hansen e Mladenovic (1997), para resolver problemas combinatoriais explorando sistematicamente a idéia de mudança de vizinhança dentro de um algoritmo de busca local, centrando a busca em torno de uma partição até que uma

solução melhor seja encontrada. Dessa forma, é possível ver o *VNS* como um processo de otimização com uma rotina de perturbação aleatória. As metodologias *K-Means, H-Means*, Busca Tabu e *VNS* são estudadas neste trabalho.

O capítulo seguinte descreve as metodologias estudadas, conceitos fundamentais utilizados pela Busca Tabu, *VNS* e as heurísticas utilizadas e modificadas.

Capítulo 2

Metodologia

Este capítulo apresenta o modelo matemático para o problema *MSSC*, os métodos não-hierárquicos do estudo, *K-Means, H-Means*, a versão do método Busca Tabu e a *VNS* . Dessa forma, na seção 2.1 é definido o problema *MSSC*, na seção 2.2 a metodologia *K-Means*, na seção 2.3 a metodologia *H-means*, na seção 2.4 a metodologia *HBaseTabu*, na seção 2.5 a metodologia *VNS* e na seção 2.6 as modificações computacionais introduzidas nas metodologias *H-Means* e Busca Tabu.

2.1 Modelo matemático

A distância entre dois clusters C_i e C_j $(i \neq j)$ é aqui definida em termos da dissimilaridade entre os centroides.

Ao longo desse trabalho, consideramos os métodos de agrupamento não-hierárquicos com o objetivo de agrupar os elementos de $\mathbf{X}$ em M subconjuntos disjuntos, de tal forma que a distância entre elementos do mesmo subconjunto seja mínima, enquanto a distância entre elementos de subconjuntos distintos seja máxima.

O problema de minimizar a soma dos quadrados das discrepâncias interna dos clusters *(MSSC)* consiste em encontrar uma partição P_M de $\mathbf{X}$ em M subconjuntos disjuntos C_i tais que a soma de quadrados dos desvios de cada elemento $\mathbf{x}_l \in C_i$ para o seu centroide $\overline{\mathbf{x}}_i$ seja mínima. Seja $\mathcal{P}_M$ o conjunto de todas as partições de $\mathbf{X}$ em M subconjuntos. Então, a *MSSC (Minimum Sum of Squares Clustering)* é dada por:

$$\text{MSSC} := \min_{P_M \in \mathcal{P}_M} \sum_{i=1}^{M} \sum_{\mathbf{x}_l \in C_i} ||\mathbf{x}_l - \overline{\mathbf{x}}_i||^2, \tag{2.1}$$

em que $||.||$ denota a norma da distância euclidiana.

2.2 Metodologia K-Means

MacQueen (1967) sugeriu o termo *K-Means* para descrever um algoritmo que atribui cada elemento ao *cluster* com o centroide mais próximo. Basicamente, o algoritmo é constituído pelas três etapas descritas a seguir.

Passo 1: Sejam C_i $\{i = 1, \cdots, M\}$, uma partição inicial (aleatória ou não) do conjunto $\mathbf{X}$ e $\overline{\mathbf{x}}_i$ seu centroide correspondente.

Passo 2: Atribuir cada elemento (um a um) para a centroide mais próximo e recalcular o centroide para o grupo que recebe o novo indivíduo e para o conjunto que perde o elemento.

Suponha que o elemento $\mathbf{x}_j$ que pertence ao *cluster* C_l é transferido para outro *cluster* C_i $(l \neq i)$. Johnson e Wichern (2002) apresentam que os centroides desses novos grupos podem ser obtidas a partir das seguintes expressões

$$\overline{\mathbf{x}}_l = \frac{n_l \overline{\mathbf{x}}_l - \mathbf{x}_j}{n_l - 1} \quad e \quad \overline{\mathbf{x}}_i = \frac{n_i \overline{\mathbf{x}}_i + \mathbf{x}_j}{n_i + 1} \tag{2.2}$$

onde $n_i = |C_i|$ e $n_l = |C_l|$. A mudança no valor da função objetivo causada por este movimento é

$$v_{ij} = \frac{n_i}{n_i + 1} ||\overline{\mathbf{x}}_i - \mathbf{x}_j||^2 - \frac{n_l}{n_l - 1} ||\overline{\mathbf{x}}_l - \mathbf{x}_j||^2.$$

Tais mudanças são computadas para todos os possíveis remanejamentos. Se eles são não-negativos a heurística para com uma partição localmente mínima. Caso contrário, a reatribuição que mais reduz o valor da função objetivo é executada.

Passo 3: Repita o *Passo 2* até não haver mais reatribuições.

2.3 Metodologia H-Means

Uma partição inicial $\{C_1, C_2, \cdots, C_M\}$ é escolhida aleatoriamente e as centroides $\overline{\mathbf{x}}_1, \cdots, \overline{\mathbf{x}}_M$ são calculadas. A heurística *H-Means* pode parar em uma solução degenerada, ou seja, com uma partição tendo menos de M *clusters* não-vazios. Neste caso, será necessária a consideração de um Passo adicional no algoritmo. A seguir é descrito o algoritmo.

Passo 1 (Inicialização): Sejam C_i $\{i = 1, \cdots, M\}$, uma partição inicial do conjunto $\mathbf{X}$ e $\overline{\mathbf{x}}_i$ seus centroides correspondentes.

Passo 2 (Atribuição): Atribuir (Alocar) cada elemento $\mathbf{x}_j$ $\{j = 1, \cdots, N\}$, para o centroide mais próximo $\overline{\mathbf{x}}_i$ $\{i = 1, \cdots, M\}$.

Passo 3 (Teste de otimalidade local): Se não houver alteração nas alocações, uma partição ótima local é encontrada. Parar.

Passo 4 (Atualização): Atualize o centroide de cada *cluster* e volte ao *Passo*

2.

Caso degenerado

Passo 3' (Teste de otimalidade local): Se existir mudança na atribuição ir para o *Passo 4*. Caso contrário, a partição local ótima é encontrada. Se for adequado, pare. Se possuir t grupos degenerados (vazios), selecione os t elementos mais distantes de suas centroides e inserir na solução como grupos de único elemento e vá para *Passo 4*.

2.4 Metodologia Busca Tabu

A metodologia Busca Tabu é uma heurística que busca explorar regiões distintas do espaço de soluções do problema com o objetivo de obter soluções que não sejam apenas mínimos locais. Para tanto, introduz conceitos e estratégias de busca tais como *Diversificação, Intensificação, Soluções-Elite* e *Lista-Tabu*, dentre outras (ver Glover e Laguna, 2002). Duas muito importantes estratégias da Busca Tabu são a *Diversificação* e a *Intensificação*. A primeira, como o próprio nome define, procura explorar distintas regiões do espaço de soluções do problema, guardando na memória soluções com características especiais encontradas, que são *Soluções-Elite*; a segunda, também como o próprio nome intui, intensifica a busca por melhores soluções na vizinhança de algumas (ou todas) das *Soluções-Elite*.

Na *Fase da Diversificação* a busca por regiões distintas dá-se através de perturbações aleatórias realizadas em componentes de determinada solução que possui atrativos, independentemente de ser a melhor solução encontrada até o momento. Já na *Fase da Intensificação* a exploração da vizinhança de uma *Solução-Elite* dá-se com maior rigor e critérios, podendo-se utilizar metodologias já bastante conhecidas e eficientes para a realização de uma busca local. Como a geração de soluções, na primeira estratégia, é feita de forma exaustiva, uma lista de movimentos já realizados deve ser construída de maneira a se evitar ciclagem (*looping*) do procedimento computacional. Esta lista de movimentos proibitivos é denominada por Lista Tabu, a qual possui tamanho limitado a ser definido como parâmetro da metodologia.

A subseção seguinte apresenta o procedimento algorítmico da heurística utilizada neste trabalho.

2.4.1 Algoritmo HBaseTabu

O algoritmo da heurística Base Tabu é denominado neste estudo por *HBase-Tabu* e é baseado no algoritmo apresentado por Al-Sultan (1995). Ele possui a seguinte

descrição: Sejam A_t, A_c e A_b partições em $\mathcal{P}_M$ (definido na seção 2.1) denotadas por partição teste, partição atual (ou corrente) e melhor partição, respectivamente, e J_t, J_c e J_b seus respectivos valores de *MSSC*.

Passo 1: Seja A_c uma partição arbirtrária e J_c o valor de *MSSC* correspondente descrito na seção (2.1).

Faça $A_b = A_c$ e $J_b = J_c$, e selecione os valores para os seguintes parâmetros: NTS, número de soluções geradas na *Fase de Diversificação* (ou número de perturbações) e ITERMAX, o número máximo de iterações. Seja k = 1, e vá para o Passo 2.

Passo 2: **(Fase da Diversificação)** Seja $N_k(A_c)$ o conjunto das vizinhanças de A_c com k elementos distribuídos em grupos distintos de A_c. Tomando-se que $\mathcal{A}_t^k \in N_k(A_c), k = 1, \cdots, NTS$, gere NTS soluções testes $A_t^1, A_t^2, \cdots, A_t^{NTS}$ e avalie as funções objetivo correspondentes $J_t^1, J_t^2, \cdots, J_t^{NTS}$ onde A_t^i é obtida de A_c trocando-se aleatoriamente a alocação de i elementos para i grupos, $i = 1, 2, \cdots, NTS$. Vá para o passo 3.

Passo 3: De forma crescente, ordene $J_t^1, J_t^2, J_t^3, \cdots, J_t^{NTS}$ e denote as por $J_t^{[1]}, J_t^{[2]}, \cdots, J_t^{NTS}$.

Se $J_t^{[1]} < J_b$, então faça $A_b = A_t^{[1]}$ e $J_b = J_t^{[1]}$ e realize uma busca intensiva aplicando o algoritmo *H-Means2* (ver seção 2.6.2) tendo como solução inicial a partição A_b. Faça $k = k + 1$ e volte para o Passo 2.

Caso contrário seja $A_c = A_t^{[1]}$, $J_c = J_t^{[1]}$ *Solução-Elite*. Faça $k = k + 1$. Se $k > ITERMAX$, vá para o Passo 4; caso contrário, volte para o Passo 2.

Passo 4: **(Fase da Intensificação)**

Sejam $A_s^1, A_s^2, A_s^3, \cdots, A_s^{INT}$ *Soluções-Elite* com valores de *MSSC* iguais a $J_s^1, J_s^2, \cdots, J_s^{INT}$, respectivamente. Seja J_s^{MIN} o valor mínimo de $J_s^1, J_s^2, \cdots, J_s^{INT}$, e A_s^{MIN} a partição correspondente. Aplicar o algoritmo *H-Means2* com A_s^{MIN} como partição inicial.

Passo 5: Seja A_f e J_f a solução obtida no passo anterior. Se $J_f < J_b$, então $J_f = J_b$ e $A_f = A_b$. A_b é a melhor solução encontrada e J_b e o melhor valor da função objetivo correspondente. Caso contrário, J_b é a melhor solução.

Repetir os Passos 4 e 5 para no máximo INT vezes.

Note que, no Passo 3, uma *Intensificação* é realizada se uma solução encontrada for melhor que o mínimo até o momento. Se a Intensificação não acontece, então a melhor solução gerada na *Fase de Diversificação* (Passo 2) é selecionada e guardada como *Solução-Elite*.

A fase de maior *Intensificação* do algoritmo compreende os Passos 4 e 5, e nela as regiões eleitas são aquelas que são vizinhanças das *Soluções-Elite* com menor *MSSC*.

2.5 Metodologia VNS

Esta Subseção apresenta a metaheurística *VNS*. Como na metodologia anterior, primeiramente considere $P_M \in \mathcal{P}_M$ uma partição qualquer dos dados $\mathbf{X}$. Uma vizinhança de P_M é representada por $\mathcal{N}_k(P_M)$, em que o parâmetro k indica em quantos elementos a vizinhança se distingue da partição P_M. Note que todos os pontos de $N_1(P_M)$ correspondem a vizinhança de P_M que é usada na heurística K-Means.

Como condição de parada dessa metaheurística é possível escolher entre tempo máximo permitido (t_{max}), número máximo de iterações ou número máximo de iterações entre duas melhorias. O algoritmo é descrito a seguir.

Passo 1 (Inicialização): Sejam $P_M = \{C_1, C_2, \cdots, C_M\}$ e f_{opt} a partição inicial do conjunto $\mathbf{X}$ e a função objetivo atual, respectivamente. Escolha alguma condição de parada e um valor para um parâmetro k_{max}.

Passo 2 (Finalização): Se a condição de parada for atendida, pare.

Passo 3 (Primeira Vizinhança): Seja $k = 1$.

Passo 4 (Circuito Interno): Se $k > k_{max}$, volte para o *Passo 2*.

Passo 5 (Perturbação): Desenhe aleatoriamente uma partição de $\mathcal{N}_k(P_M)$, i.e., redesignar quaisquer k elementos de $\mathbf{X}$ para outros clusters diferentes do atual. Denotar a partição obtida por P'_M.

Passo 6 (Busca local): Aplique uma heurística de busca local (considerando P'_M como partição inicial). Denotar a solução resultante e valor da função objetivo por P''_M e f'', respectivamente.

Passo 7 (Mova ou não): Se $f'' < f_{opt}$, então centralize a busca em torno do melhor solução encontrada ($f_{opt} = f''$ e $P_M = P''_M$) e vá para o *Passo 3*. Caso contrário, definir $k = k + 1$ e retornar ao *Passo 4*.

Nota-se que, diferentemente da Busca Tabu, a *VNS* realiza uma busca local a cada pertubação realizada (Passos 5 e 6) e realiza mudança assim que encontra uma solução melhor que a atual (Passo 7).

2.6 Modificações de H-Means e Busca Tabu

Como mencionado em seções anteriores, as metodologias *H-Means* e Busca Tabu foram implementadas, em linguagem compatível ao Sistema Computacional Estatístico R, por Quintino (2015) e Távora (2015), respectivamente e seus algoritmos denominados por *H-Means* e *HBaseTabu*.

Com o propósito de melhor adequar *HBaseTabu* às estratégias da metodologia Busca Tabu, modificações no código computacional de *HBaseTabu* foram realizadas e são descritas na seção 2.6.1. Além disso, algumas modificações foram introduzidas no código *H-Means* visando melhorar adequação ao sistema utilizado.

2.6.1 Algoritmo HBaseTabu2

Com o intuito de melhor explorar as *Estratégias da Diversificação* e *Intensificação* da Metodologia Busca Tabu, criou-se, para o Passo 1, um novo parâmetro, e os Passos 4 e 5 foram reprogramados. Dessa forma, lê-se:

Passo 1: O parâmetro NTS passa a ser o número máximo de perturbações realizadas na solução atual. O parâmetro NTS1 é introduzido para determinar o número mínimo de perturbações.

Passos 4 e 5 passaram a escolher, para a *Intensificação*, a *Solução Elite* com o menor valor de *MSSC* e a possibilitar a repetição desta Fase em um número de até INT vezes (ver *Intensificação em Soluções-Elite* no Apêndice B).

Nota-se que o Passo 1, com o novo parâmetro, possibilita gerar soluções mais distantes da Solução atual, enquanto que a repetição dos Passos 4 e 5 possibilitam uma *Intensificação* mais seletiva das *Soluções-Elite*.

2.6.2 Algoritmo H-Means2

O programa em linguagem R da heurística *H-Means*, elaborado por Quintino (2015), foi adaptado ao sistema R com o intuito de se minimizar seu gasto computacional. Para tanto, alguns comandos *for* foram substituídos por *function*, como, por exemplo, o que segue abaixo:

```
Substituição do comando ''for'':
for (elem1 in 1:n){ dados[elem1,r+1] <- (dist(rbind(dados
[elem1,1:(r-1)],
centro[(dados[elem1,r]), ]), method = "euclidean"))^2 } #Parte 1

Pelo comando ''distancias'':
distancias<-function(elem1){dados[elem1,r+1]<- (dist(rbind
(dados[elem1,1:(r-
1)],centro[(dados[elem1,r]),]),method="euclidean"))^2}
sapply(1:nrow(dados),distancias)
```

No próximo capítulo serão apresentados os resultados computacionais obtidos com as modificações de *H-Means* e Busca Tabu.

Capítulo 3

Resultados e Discussão

Este capítulo apresenta os resultados computacionais obtidos e a análise comparativa entre os métodos descritos no capítulo anterior. Para tanto, utilizou-se o algoritmo *K-Means* disponível no sistema R. Os algoritmos *H-Means* e *VNS* foram implementados nesse mesmo sistema, e disponibilizados por Quintino F. (2015). O *HBaseTabu* foi implementado e disponibilizado por Távora (2015). Os algoritmos *H-Means2* e *HBaseTabu2* referem-se à segunda versão de *H-Means* e *HBaseTabu* com modificações implementadas pelos próprios autores deste trabalho, como mencionados na seção anterior. Suas implementações foram realizadas em linguagem compatível à do Sistema de Computação Estatística R. As implementações do algoritmo *HBaseTabu2* e *H-Means2* estão apresentadas nos anexos A e B. Os resultados desse estudo foram obtidos através de testes com bancos de dados do próprio Sistema de Computação Estatística R. Esses bancos de dados são: USArrests e Íris. Todos os métodos mencionados na seção anterior foram testados e os resultados obtidos são apresentados com detalhes nas próximas subseções.

3.1 Testes com USArrests

Este banco de dados consiste em cinqüenta observações, as quais são referentes aos estados dos Estados Unidos em 1973 para cada 100.000 habitantes. Cada observação possui quatro variáveis, que se referem ao número de assassinato, assalto, estupro e população urbana para cada 100.000 habitantes. Todas as variáveis consideradas são quantitativas.

Com esse banco de dados, testou-se todos os métodos do estudo. Em um primeiro teste, realizou-se 10 iterações para cada algoritmo. Nos métodos que necessitam de perturbações na Fase de Diversificação, como o *HBaseTabu2* foram realizadas 20 pertubações. Para isso, na parte da intensificação, foram escolhidas 8 soluções elites para serem exploradas. Os resultados obtidos são apresentados na Tabela 3.1, e são

aqueles referentes ao melhor resultado encontrado dentre oito inicializações realizadas

Tabela 3.1: Desvio do K-Means em relação aos outros métodos para 10 iterações e 20 pertubações para HBaseTabu2

		Desvios		
M	K-Means	H-Means2	HBaseTabu2	VNS
2	96399.03	0.00	0.00	0.00
3	47964.27	0.00	0.00	0.00
4	37652.66	- 663.06	-2924.03	-2924.03
5	24417.02	3823.21	87.93	82.37
6	22290.84	- 3522.84	-4182.89	-3440.47
7	16563.67	1200	343.85	-431.59
8	13259.15	1200	1516.4	1065.34
Erro Médio	0.00	254.66	- 644.8425	-706.0475

A Tabela 3.1 mostra na segunda coluna os valores de *MSSC* obtidos pelo método *K-Means*, que é a heurística base para as comparações. Os valores das colunas seguintes se referem ao desvio dessa heurística em relação ao H-Means2, *HBaseTabu2* e *VNS*, respectivamente. As comparações foram feitas por esses métodos sendo realizadas 10 iterações e 20 pertubações para cada caso. Com isso, observou-se que os métodos *HBaseTabu2* e *VNS* obtiveram melhoras significativas, tendo para grupos de tamanho M=4, M=6 e M=7 valores melhores que o *K-Means*.

Comparando os algoritmos *HBaseTabu2* e *H-Means2*, pode-se notar que o desempenho do primeiro foi melhor para a maioria dos números de agrupamentos. Isso ocorre já que o erro médio dessa heurística foi inferior ao da segunda. Por outro lado, comparando o *HBaseTabu2* com *VNS*, é possível notar equilíbrio entre esses métodos. Ambos obtiveram melhoras em seus desempenhos finais, com pequenas diferenças entre os valores de *MSSC* e com bons resultados para cada número de agrupamento de dados. Particularmente *HBaseTabu2* mostrou-se superior a *VNS* para M=6. Para os demais números de agrupamentos, a *VNS* obteve resultados iguais ou melhores que *HBaseTabu2*

Em geral, os algoritmos implementados no Sistema de Computação R obtiveram bons resultados quando comparados ao *K-Means*, principalmente à medida que o número de grupos aumenta, melhorando seus resultados e diminuindo o erro médio de seus métodos. Dentre essas heurísticas, o *HBaseTabu2* obteve esse desempenho, já que utiliza os procedimentos da *Intensificação* e *Diversificação* para buscar valores em diferentes regiões e assim conseguir otimizá-los. Para melhor observar esse comportamento, a Tabela 3.2 apresenta resultados obtidos pelo *HBaseTabu2* antes ($J_b$inicial) e depois ($J_b$final) da *Fase de Intensificação*.

Tabela 3.2: Comparação do valor de *MSSC* antes e ao final da *Fase de Intensificação* do Método *HBaseTabu* para 20 perturbações, 10 iterações, USArrests

M	$J_b$inicial	$J_b$final
2	96399.03	96399.03
3	47964.27	47964.27
4	36989.6	34728.63
5	28240.23	24504.95
6	26472.24	18107.95
7	25800.21	16907.52
8	14775.55	14775.55

Como já mencionado na seção 2.4, a *Intensificação* foi implementada no algoritmo *HBaseTabu2* com o intuito de procurar diminuir mais o valor da função objetivo. Logo, o $J_b$final representa o valor da *MSSC* após o final dessa intensificação. Observando a Tabela 3.2, percebe-se que os valores, tanto inicial como final, são iguais para o número de grupos M=2 e M=3. Porém, essa igualdade não permanece para os demais grupos. Isso ocorre, pois esse método retorna a regiões mais atraentes fazendo uma busca mais profunda nesta região, e com isso consegue obter resultados melhores.

Nota-se que para 20 perturbações a *Fase da Intensificação* foi efetiva. Ao observar os valores de $J_b$inicial e $J_b$final, percebe-se que esses valores sofrem uma diminuição. A partir do número de grupos igual a 4, os valores de *MSSC* são melhores ao final, o que sugere a importância da *Fase da Diversificação* e *Intensificação* para a metodologia. Em um segundo teste comparou-se os métodos *K-Means, HBaseTabu* e *HBaseTabu2*. Para cada algoritmo realizou-se 15 iterações e 10 pertubações para *HBaseTabu* e *HBaseTabu2*. Para isso, na parte da *Intensificação*, foram escolhidas 8 soluções elites para serem exploradas, como mostra a Tabela 3.3.

A Tabela 3.3 mostra na segunda coluna, os valores de *MSSC* obtidos pelo método *K-Means*, que é a heurística base para as comparações. Já os valores mostrados nas colunas seguintes se referem ao desvio dessa heurística em relação ao *HBaseTabu* e *HBaseTabu2*, respectivamente.

Com isso, observa-se que o método *HBaseTabu2* obteve melhoras significativas para M=4 e M=6 com valores de *MSSC* melhores que o *K-Means*. Ao comparar com *HBaseTabu*, o *HBaseTabu2* apresenta melhora dos valores de *MSSC* para todos os números de grupos, tendo seu erro médio expressivamente menor que o de *HBaseTabu*. Esta melhora pode ser explicada devido às modificações implementadas em *HBaseTabu*, mencionadas na seção 2.6.1.

Tabela 3.3: Desvio do *K-Means* em relação aos outros métodos para 15 iterações e 10 pertubações

		Desvios	
M	K-Means	HBaseTabu	HBaseTabu2
2	96399.03	0.00	0.00
3	47964.27	0.00	0.00
4	37652.66	3823.21	- 2924.03
5	24417.02	3823.21	87.93
6	22290.84	- 3179	- 3522.84
7	16563.67	1708.11	1396.28
8	13259.15	4005.64	2621.83
Erro Médio	0.00	1454.45	- 334.285

3.2 Testes para a Íris de Fisher

Em um segundo teste foi utilizado o conjunto de dados Iris de Fisher que é composto por 150 observações, as quais são referentes a três espécies de plantas, que são a setosa, versicolor e virgínica. Cada observação ou elemento tem cinco variáveis. Dentre as variáveis, quatro são classificadas como variáveis quantitativas e uma qualitativa. A variável qualitativa é a variável que classifica as espécies, e não foi considerada na análise de *clusters* em estudo. As quatro variáveis quantitativas são: Largura da pétala, Largura da sépala, Comprimento da pétala e Comprimento da sépala. Com esse banco de dados, testes computacionais foram realizados com os métodos em estudo neste trabalho.

Em um primeiro teste, o método *HBaseTabu2* foi testado, utilizando-se na *Fase da Diversificação*, 30 perturbações e foi comparado com os métodos *K-Means, H-Means2* e *VNS*. Todos os algoritmos foram testados com número de iterações igual a 10. A Tabela 3.4 mostra os resultados, que são aqueles referentes ao melhor resultado encontrado dentre oito inicializações realizadas

Os valores ótimos da função objetivo, *MSSC*, foram extraídos do artigo dos autores Hansen e Mladenovic (2001). Logo, com esses valores ótimos, foi feita a comparação pelos desvios e os erros médios de cada método. Na Tabela 3.4 é possível observar que os valores de *MSSC* obtidos pela heurística *H-Means2* são em média melhores que *K-Means* para 10 iterações, uma vez que a cada iteração de *K-Means* é realizada apenas uma realocação que minimiza a função objetivo enquanto *H-Means* realiza simultaneamente todas as realocações que podem minimizar esta função. Nota-se que *H-Means2* pode apresentar melhores resultados que *K-Means* para número de iterações pequeno.

Tabela 3.4: Desvio dos métodos em relação ao valor ótimo com 30 perturbações e 10 iterações

		Desvios			
M	Ótimo	K-Means	H-Means2	HBaseTabu2	VNS
2	152,347	0.00	0.00	0.00	0.00
3	78,8525	0.00	0,07	0.00	0.00
4	57,2284	0,04	0,03	0.00	0,04
5	46,4461	3,38	6,55	0.00	0,03
6	39,0399	2,66	3,15	3,12	0,02
7	34,2982	12,76	10,2	3,1	0,02
8	29,9889	5,82	3,81	2,56	0,63
9	27,786	6,3	1,07	0,49	3,09
10	25,834	0,84	1,43	0,79	0,84
Erro médio		3,18	2,63	1,11	0,52

Os resultados da *HBaseTabu2* foram em média melhores que os obtidos por *K-Means* e *H-Means2*, e a *VNS* foi a que obteve o menor erro médio. Pelos desvios, percebe-se que os valores da função objetivo, ou *MSSC*, são os mesmos para todos os métodos para o número de grupos igual a 2. Porém, percebe-se que esse comportamento de igualdade não é mais o mesmo para os demais valores de M. Todas as metodologias tiveram solução ótima para M=3 com exceção de *H-Means2*. A heurística *HBaseTabu2* obteve solução ótima para M=4 e M=5, ao contrário dos demais que não conseguiram chegar a esse resultado. Para M=6, M=7 e M=8 a *VNS* obteve os melhores resultados com valores bem próximos ao ótimo. Para M=9 e M=10, *HBaseTabu2* obteve os melhores resultados.

Com o propósito de melhor explorar a *Estratégia da Diversificação* da heurística *HBaseTabu2*, realizou-se testes com distintas versões desta heurística. Para tanto, todos os algoritmos foram testados com número de iterações igual a 15. Para *HBaseTabu2* utilizou-se 30 perturbações, para a *HBaseTabu2+* foram realizadas de 30 a 50 perturbações e para a *HBaseTabu2++* de 50 a 100 perturbações. A Tabela 3.5 apresenta os resultados obtidos pelas versões implementadas, incluindo resultados de *HBaseTabu* (Távora, 2015) para 15 iterações e 30 perturbações.

Para M=2 até M=7 os métodos obtiveram os mesmos resultados com exceção de *HBaseTabu* que de M=5 até M=7 obteve valores maiores de *MSSC*. De M=8 a M=10 a *HBaseTabu* obteve resultados não muito próximos em relação ao valor ótimo. Para M=8 *HBaseTabu2++* e *HBaseTabu2+* obtiveram melhores resultados com relação a *HBaseTabu2*, sendo que a primeira foi a que teve o menor valor. Para a M=9 foi a vez de *HBaseTabu2+* ter o melhor resultado e as outras com valores relativamente

Tabela 3.5: Desvio da Íris em relação ao valor ótimo com diferentes perturbações, com número de iterações e intensificações igual a 15, com exceção de HBaseTabu

		Desvios			
M	Ótimo	HBaseTabu	HBaseTabu2	HBaseTabu2+	HBaseTabu2++
2	152.347	0.00	0.00	0.00	0.00
3	78.8525	0.00	0.00	0.00	0.00
4	57.2284	0.00	0.00	0.00	0.00
5	46.4461	6.46	0.00	0.00	0.00
6	39.0399	3.37	3.12	3.12	3.12
7	34.2982	9.4	3.1	3.1	3.1
8	29.9889	5.77	2.56	0.44	0.39
9	27.786	7.97	0.49	0.42	0.57
10	25.834	7.97	0.79	0.96	0.96
Erro médio		4.55	1.11	0.89	0.90

próximos. E por último, para M=10, *HBaseTabu2* teve o melhor resultado dentre as quatro heurísticas. O menor erro médio foi da *HBaseTabu2+* seguido da *HBaseTabu2++* com valor bem próximo. A *HBaseTabu2* obteve o terceiro menor erro médio, mas não muito distante dos dois menores e por fim a *HBaseTabu* obteve o maior erro médio.

Para verificar o desempenho da *HbaseTabu2* com relação às estratégias da intensificação e diversificação implementadas, a Tabela 3.6 apresenta resultados obtidos dos valores de *MSSC* antes ($J_b$inicial) e depois ($J_b$final) da *Fase de Intensificação* para os algoritmos *HBaseTabu2, HBaseTabu2+, HBaseTabu2++.*

Tabela 3.6: Comparação do valor de *MSSC* antes e ao final da *Fase de Intensificação* dos algoritmos *HBaseTabu2, HBaseTabu2+* e HBaseTabu2++, 15 iterações.

M	HBaseTabu2		HBaseTabu2+		HBaseTabu2++	
	$J_b$inicial	$J_b$final	$J_b$inicial	$J_b$final	$J_b$inicial	$J_b$final
2	152.348	152.348	152.348	152.348	152.348	152.348
3	78.8525	78.8525	78.85567	78.8525	78.85567	78.8525
4	57.25601	57.22847	57.25601	57.22847	57.25601	57.22847
5	52.99335	49.86225	52.99335	46.44618	52.99335	46.44618
6	41.87319	41.70591	42.4247	42.16294	51.80504	42.16294
7	40.34128	40.34128	43.72647	37.39662	40.39461	37.39514
8	35.81144	35.75025	30.33777	30.12946	30.3812	30.37886
9	32.19166	32.10937	28.53636	28.20223	34.35487	28.1005
10	33.66422	33.21159	36.24578	26.79218	26.80837	26.79218

Em primeiro lugar, nota-se que J_b inicial difere nos testes devido ao fato de

a solução inicial ser gerada aleatoriamente. Independentemente deste fato, observa-se, pelos resultados apresentados, que para *HBaseTabu2* de M=4 até M=10, com exceção de M=7, houve uma pequena melhora no valor do *MSSC*, para os outros casos o valor manteve-se. Para o algoritmo *HBaseTabu2+* houve uma melhora considerável a partir de M=3 no valor da função objetivo. Os resultados obtidos pela *HBaseTabu2++* mostram que a *Fase de Intensificação* melhora os valores de *MSSC* para M=5, 6, 7 e 9.

3.3 Análise Exploratória do USArrests

Os bancos de dados utilizados para os testes deste trabalho foram a Íris de Fisher e o USArrests. Essas informações são acessíveis para qualquer usuário do sistema operacional R. O USArrests é um banco de dados interessante, pois traz consigo informações interessantes de serem estudadas, como: índice de assassinato, assalto, estupro. Essas informações são referentes aos estados dos Estados Unidos em 1973 para cada 100.000 habitantes, e nos revelam características importantes desses crimes nesse país como segue a Tabela 3.7 e Tabela 3.8

Tabela 3.7: Valores da média e mediana para as variáveis de crime

	Assassinato	Assalto	Estupro
Média	7.788	170.76	21.232
Mediana	7.250	159.00	20.100

Tabela 3.8: Quartis referentes aos tipos de crime

	Assassinato	Assalto	Estupro
0%	0.800	45	7.300
25%	4.075	109	15.075
50%	7.250	159	20.100
75%	11.250	249	26.175
100%	17.400	337	46.000

Essas tabelas e a Figura 3.1 mostram o comportamento das variáveis nesse banco de dados. Observa-se que o índice de assaltos (*assault*) é significativamente superior as demais, tendo no estado da Carolina do Norte os maiores índices. Isso mostra a realidade vivida nos estados dos Estados Unidos no ano em que esses dados foram coletados. Outro aspecto interessante, é a presença de um outlier somente nos dados referentes ao estupro (*Rape*), mostrando que essas variáveis seguem um mesmo comportamento tendo poucos pontos discrepantes. Esse tipo de crime teve uma maior

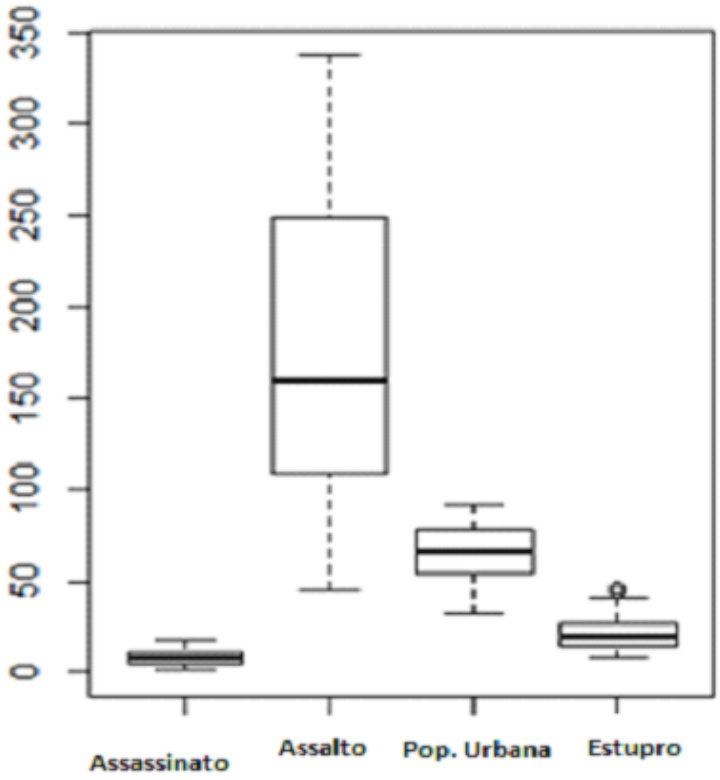


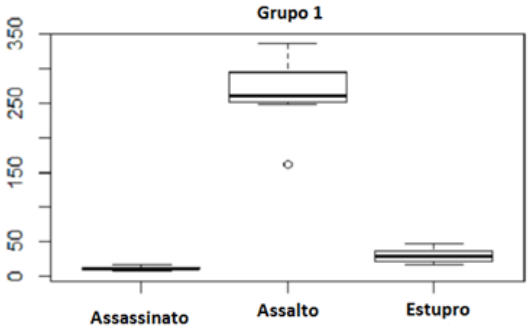
Figura 3.1: Boxplot das variaveis de crime

incidência no estado Nevada. E o estado do Norte de Dakota as maiores infringências de assassinato. A Tabela 3.9 e a Figura 3.1 mostram a análise de correlação entre esses crimes.

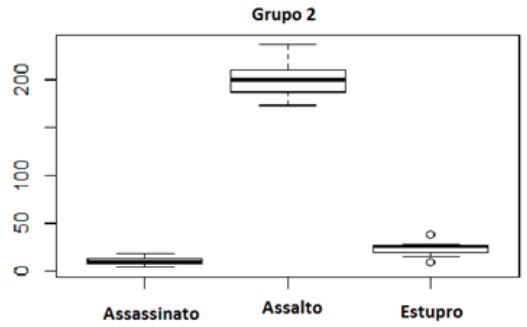
Tabela 3.9: Correlação entre as variaveis de crime

	Assassinato	Assalto	Estupro
Assassinato	1.0000000	0.8018733	0.5635788
Assalto	0.8018733	1.0000000	0.6652412
Estupro	0.5635788	0.6652412	1.0000000

A Tabela 3.9 mostra os valores para a correlação entre os crimes estudados. Observa-se que existe uma forte correlação entre as variáveis assalto e assassinato, mostrando que a maioria dos assassinatos são oriundos de um assalto. As outras relações entre as variáveis do crime geram resultados de leve relações entre elas, isso demonstra que mesmo que não tenham uma grande ligação essas modalidades de crime tem relações entre si.



	Assassinato	Assalto	Estupro
0%	6.80	161.00	15.60
25%	10.10	252.50	22.65
50%	11.35	261.00	29.40
75%	12.80	291.75	34.35
100%	16.10	337.00	46.00



	Assassinato	Assalto	Estupro
0%	3.4	174	8.3
25%	7.9	188	19.5
50%	9.0	201	25.5
75%	13.2	211	26.9
100%	17.4	238	38.7

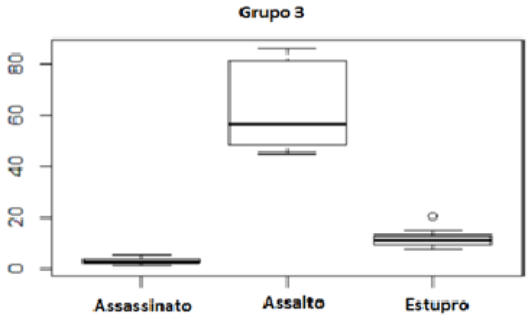
Figura 3.2: Boxplot e Esquema dos Cinco Números para as variáveis índice de assassinato, assalto e estupro dos grupos 1 e 2

3.3.1 Análise do USArrests para Quatro Agrupamentos

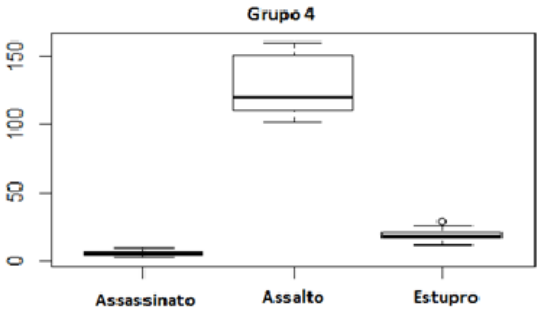
Uma análise do *cluster* de tamanho M=4 do banco de dados USArrests foi realizada com o intuito de inferir sobre a melhora obtida pelos algoritmos, como mostra a Tabela 3.1. Para isso, foram gerados boxplots e quartis referentes as variáveis *assassinato*, *assalto* e *estupro* desse grupo como mostra a Figura 3.2 e 3.3.

A Figura 3.2 mostra como os elementos do banco de dados ficaram distribuídos nesses dois primeiros grupos. O grupo 1 teve valor mínimo de 6.80 para a variável *Assassinato*, 161.00 para *Assalto* e 15.60 para *Estupro*. Já os valores máximos para cada variável foram 16.10, 337.00 e 46.00 respectivamente. Esse grupo tem um valor discrepante para a variável *Assalto* que é o estado de Wyoming, que possui um menor índice desse crime no ano 1973. O grupo 2 gerou valores mínimos para as variáveis *Assassinato*, *Assalto* e *Estupro* de 3.4, 174.00 e 8.3, respectivamente. Os valores máximos desse grupo foram 17.4, 238.00 e 38.7 como mostra a Figura 3.2.

A Figura 3.3 mostra como os elementos do banco de dados ficaram distribuídos nesses dois últimos grupos. O grupo 3 teve valor mínimo de 0.800 para a variável *Assassinato*, 45.00 para *Assalto* e 7.300 para *Estupro*. Já os valores máximos para cada variável foram 5.700, 86.00 e 20.200 respectivamente. O grupo 4 gerou valores mínimos para as variáveis *Assassinato*, *Assalto* e *Estupro* de 2.6, 102.00 e 11.1,



	Assassinato	Assalto	Estupro
0%	0.800	45.00	7.300
25%	2.125	49.25	9.350
50%	2.400	56.50	11.000
75%	3.525	78.75	12.425
100%	5.700	86.00	20.200



	Assassinato	Assalto	Estupro
0%	2.6	102	11.1
25%	4.3	110	16.3
50%	6.0	120	18.0
75%	7.2	151	21.0
100%	9.7	161	29.3

Figura 3.3: Boxplot e Esquema dos Cinco Números para as variáveis índice de assassinato, assalto e estupro dos grupos 3 e 4

respectivamente. Os valores máximos desse grupo foram 9.7, 161.00 e 29.3 como mostra a Figura 3.3. Analisando os valores e gráficos, observa-se que as variáveis *Assassinato* e *Estupro* têm uma pequena variação em seus valores e a variável *Assalto* tem uma variação significativa. Nota-se que os quatro grupos apresentam índices de assalto diferentes, cujos valores variam em intervalos praticamente disjuntos com excessão do *outlier* obsevado no grupo 1. Com isso, pode-se dizer que esta variável foi determinante para a definição dos grupos em questão. E que o resultado obtido pelo algoritmo *HBaseTabu2* mostra-se coerente e eficaz. Isso mostra que é adequado esse número de grupo para uma melhor avaliação do banco de dados USArrests.

A partir das figuras apresentadas, nota-se que os valores do 1º quartil para a variável *Assalto* foram 252.50, 188.00, 49.25 e 110 para os 4 grupos respectivamente. Para os outros quartis, os valores observados foram de 56.50, 120.00, 261.00, 201.00 para o 2º quartil e 291.75, 211.00, 78.75 e 151 para o 3º quartil. Isso mostra que essa variável possui uma variabilidade maior em seus valores para cada grupo, tendo uma amplitude de 176.00 no grupo 1, 64.00 no grupo 2, 41.00 no grupo 3 e 59.00 no grupo 4. Em relação a variável *Assasinato*, os valores encontrados foram de 10.10, 7.9, 2.125, 4.3 para o 1º quartil, 11.35, 9.0, 2.400, 6.0 para o 2º quartil e 12.80, 13.2, 3.525, 7.2 para o 3º quartil. Tais informações mostram que essa variável possue valores próximos em seus grupos, tendo amplitudes de 9.3 no grupo 1, 14.0 no grupo 2, 4.9 no grupo 3

e 7.1 no grupo 4. Já a variável *Estupro* teve valores de 22.65, 19.5, 9.350, 16.3 para o 1º quartil, 29.40, 25.5, 11.000, 18.0 para o 2º quartil e 34.35, 26.9, 12.425, 21.0 para o 3º quartil. Com isso tal variável tem uma moderada variação em seus valores, tendo amplitude de 30.4 para o grupo 1, 30.4 no grupo 2, 12.9 no grupo 3 e 18.2 no grupo 4. Logo através dessas informações aqui descritas, nota-se que a variável *Assalto* possui uma maior amplitude em seus valores em relação as outras variáveis, como mostra os boxplots das figuras 3.2 e 3.3.

Capítulo 4

Considerações Finais

Este trabalho apresenta estudos comparativos entre heurísticas clássicas e recentes na resolução de problemas *MSSC*, tais como K-Means, H-Means, *Tabu Search* e *VNS*. Para tanto, mudanças na implementação da versão algorítimica existente da metodologia Busca Tabu, em ambiente do Sistema Computacional Estatístico R, foram realizadas. O algoritmo implementado teve como base o trabalho de Távora (2015), alterando o programa existente no que se refere às *Fases de Diversificação* e *Intensificação* da metodologia em questão.

Dos testes computacionais realizados, pode-se concluir que a implementação elaborada foi eficaz, apresentando bons resultados quando comparados àqueles obtidos pelas metodologias *K-Means*, *H-Means* e *VNS*. Os resultados apresentados na Tabela 3.6 indicam que as *Estratégias da Diversificação* e *Intensificação* implementadas são efetivas, ou seja, geram melhorias dos resultados metodológicos.

Particularmente, os testes realizados com o arquivo de dados USArrests mostraram que o algoritmo *HBaseTabu2* obteve melhoras siginificativas em relação ao *K-Means* nos valores de *MSSC* para números de grupos iguais a M=4 e M=6 das Tabela 3.1 e 3.3, comprovando a adequação da implementação dessa metodologia.

Análise exploratória dos resultados obtidos, demonstra a eficácia do resultado obtido para quatro agrupamentos do banco de dados USArrests. O alto índice de assaltos e suas relações com a variável *Assassinato* foram relatados na análise dos crimes ocorridos nesse ano nos Estados Unidos. Ao analisarmos o banco de dados para M=4, observou-se que a variável *Assalto* tem uma variação maior em relação as outras variáveis, sendo que este grupo possui dados mais homogêneos. Tal fato demonstra os bons resultados obtidos pelo algoritmo *HBaseTabu2*.

Nos testes realizados com o arquivo Irís de Fisher, foi necessário ampliar a busca por diversificações de regiões e o número de *Intensificações*, devido tratar-se de arquivo com maior número de elementos que o primeiro. A necessidade de ampliar a *Fase da Diversificação* e/ou *Intensificação* é uma exigência da Metodologia

Busca Tabu (TS), bastante conhecida na literatura da Área de Pesquisa Operacional. Os resultados desses testes mostraram que para M=6, M=7 e M=8, a *HBaseTabu2* alcança valores bem próximos ao ótimo como mostra a Tabela 3.4 da seção 3.2. Já ao observar a Tabela 3.6 dessa mesma seção, observa-se que os resultados obtidos pela *HBaseTabu2++* mostram que a *Fase de Intensificação* melhora os valores de *MSSC* para M=5, M=6, M=7 e M=9.

A melhora no desempenho do algoritmo, deve-se a mudanças realizadas neste trabalho. Tais alterações intensificaram a procura por valores para suas soluções elites. Notadamente, os resultados obtidos pelas versões de *HBaseTabu2* mostraram-se melhores que aqueles obtidos pela versão original *HBaseTabu*.

Os custos computacionais observados para *HBaseTabu2* são menores que os observados para *VNS*, porém maiores que os apresentados por *K-Means*. Este fato sugere que, para banco de dados maiores, o algoritmo *HBaseTabu2*, para competir com *K-Means*, deve ser programado utilizando linguagens computacionais científicas mais eficientes.

Uma consideração importante a ser feita é que os algoritmos *H-Means2* e *HBaseTabu2* são bons algoritmos, mas podem ainda ser melhorados. As modificações foram feitas e testadas para os bancos de dados USArrests e Irís de Fisher, mas seria interessante testá-las para banco de dados maiores. Os resultados assim obtidos devem gerar melhores informações para o estudo, tornando a implementação dessa metodologia uma boa ferramenta para estudos de caso com banco de dados maiores.

Referências Bibliográficas

[1] Aloise, D., Hansen, P., **Clustering: A Chapter for Handbook for Discrete and Combinatorial Optimization**, Les Cahiers du GERAD, April, (2008).

[2] Al-Sultan, K., A Tabu search approach to the clustering problem, **Pattern Recognit**, vol. 28, no. 9, pp 1443-1451, (1995).

[3] Babu, G. e Murty, M., A near-optimal initial seed value selection in K-means algorithm using a genetic algorithm, **Pattern Recognit**, Lett., vol. 14, no. 10, pp 763-769, (1993).

[4] Biagio, M. A., **A Recovering Comparative Study of Clustering Analysis**, Tendências em Matemática Aplicada e Computacional, 1, n.2, pp 303-317, (2000).

[5] Brusco, M.J., Steinley, D., **A Comparison of Heuristic Procedures for Minimum Within-Cluster Sums of Squares Partitioning**, Psychometrika, vol.72, 4, pp 583-600, (2007).

[6] Everitt, B., **Cluster Analysis**, London, Heinemann Ed. Books, (1993).

[7] França, P.M., Sousa, N.M., Pureza, V., **An Adaptive Tabu Search Algorithm for the Capacitated Clustering Problem**, Internacional Transactions in Operational Research, Volume 6, pp 655-678, (1999).

[8] Glover, F., Laguna, M., **Tabu search**, Kluwer Academic Publishers, (2002).

[9] Hansen, P., Mladenovic N., J-MEANS: a new local search heuristic for minimum sum of squares clustering, **Pattern Recognition** 34, pp 405-413, (2001).

[10] Hansen, P., Mladenovic N., **Variable Neighborhood Search**, In: Burke, E., Kendall,G.; Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques; Springer Science, N.Y., pp 211-238, (2005).

[11] Jain, A. K., Data Clustering: 50 Years Beyond K-means, **Pattern Recogn**, Lett.,31, pp 651-666, (2010).

[12] Johnson, R. A. e Wichern, D. W., **Applied Multivariate Statistical Analysis**, sexta edição, pp 671-715, (2007).

[13] Kaufman, L. e Rousseuw, P., Finding Groups in Data: **An Introduction to Cluster Analysis**, Wiley, New York, (2005).

[14] Ng, M. K. e Wong, J. C., Clustering Categorical Data Sets Using Tabu Search Techniques, **Pattern Recognition** vol. 35, pp 2783-2790, (2002).

[15] Parsha, M. K. e Pacha, S., **Recent Advances in Clustering Algorithms**: A Review,Int. J. of Conceptions on Computing and Information Technology, vol.1, Issue 1.,Nov. 2013, (2013).

[16] Quintino, F. S., **Metodologia para Agrupamento de Dados: versão VNS para Sistema R**, PIBIC 2014-2015, Trabalho apresentado no Congresso de Iniciação Cientifica, (Brasília, DF), 2015.

[17] R Core Team (2015). **R: A language and environment for statistical computing. R** Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

[18] Tavora, M. S., **Metodologia para Agrupamento de Dados - Uma versão da Busca Tabu para Sistema R** , Trabalho de Conclusão de Curso 2014-2015, (UnB, Brasília, DF), 2015

[19] MacQueen, J. B., Some methods for classication and analysis of multi-variate observations, Proceeding of the 5th **Berkeley Symposium on Mathematical Statistics and Probability**, Vol. 2, 1967, pp. 281-297.

[20] Forgy, E. W. (1965) **Cluster analysis of multivariate data: efficiency vs interpretability of classifications**. Biometrics 21.

Apêndice A

Código da versão H-Means2

Neste apêndice apresenta o código modificado em linguagem R de computação estatística da versão do Algoritmo H-Means do Capítulo 2.

```
################################################
####### Uma versao do  H-Means2 #########
################################################
H.means2 <- function (dados, M, iter.max=10, initial.part = 1){
  r <- ncol(dados)+1
  n <- nrow(dados)
  cnames <- names(dados)
  if (is.matrix(dados)){
  dados <- as.data.frame(dados)}
  f1 <- function(dados, M, r, n){
  distancias <- function(elem1){
  (dist(rbind(dados[elem1,1:(r-1)]
  ,centro[(dados[elem1,r]),])
  ,method="euclidean"))^2 }
  dados[,r+1] <- sapply(1:nrow(dados),distancias)
  vet.opt <<- tapply(dados[,r+1], dados[,r], sum)
  f.obj <<- sum(vet.opt)
  dist1 <<- dados[,r+1]
  structure(list(vet.opt2 = vet.opt,
  f.obj2 = f.obj, distancias = dist1)) }
  ########## Passo 1 ################################
  if (length(initial.part) == 1){
  dados[,r] <- as.factor(floor(runif(n,1,M+1))) }
  if (length(initial.part) > 1){
  dados[,r] <- as.factor(initial.part)  }
  centroid <- function(dados, M, r){
  dados[,r+1] <- numeric(nrow(dados))
  centro.atual <<- matrix(0,nrow = M, ncol = (r-1))
  for (cl5 in 1:M){
  if(cl5%in%dados[,r]){
```

```
coluna <- function(i){
tapply(dados[,i], dados[,r]==cl5, mean)[2] }
centro.atual[cl5,] <- sapply(1:(r-1), coluna) }
else { centro.atual[cl5,] <- rep(0,(r-1)) }}
return(centro.atual)}
########## Passo 2 ################################
passo2 <- function(dados, M, r){
dados[,r+1] <- numeric(nrow(dados))
centro <<- centroid(dados, M, r)
elemento <- function(i){
dij <- numeric(M)
posicao <- function(p0){
dist(rbind(dados[i,1:(r-1)], centro[p0,]),
method = "euclidean") }
dij <- sapply(1:M, posicao)
return(which(dij==(min(dij))))   }
dados[,r] <- as.factor(sapply(1:n, elemento))
return(dados[,r])}
#Fim passo2
#condicao 3'
condicao.3 <- function(dados, r){
retorne <- 0
ret <- 0
for (tent in 1:length(levels(dados[,r]))){
if (tent%in%dados[,r]){ ret <- ret + 1} }
if (ret == length(levels(dados[,r]))){ retorne <- 1}
return(retorne) }
#Fim condicao 3'
for (repeticao in 1:iter.max){
centro <<- centroid(dados, M, r)
if (condicao.3(dados, r) == 1){
dados[,r] <- passo2(dados, M, r) }
else {
#Passo 3'
for (cl in 1:M){
if (cl%in%dados[,r] == F){
dmax <- 0
novo <- 0
for (elem in 1:n){
d2 <- dist(rbind(dados[elem,1:(r-1)],
centro[dados[elem,r],]),
method = "euclidean")
if ( d2 > dmax ){
novo <- elem
dmax <- d2 } }
dados[,r] <- as.numeric(levels(dados[,r])[dados[,r]])
```

```
dados[novo,r] <- cl
dados[,r] <- as.factor(dados[,r])
centro <<- centroid(dados, M, r) } }}}
#Fim repeticao
otimo <- f1(dados, M, r, n)
f.opt <<- otimo$f.obj2
dimnames(centro) <- list(1:M,cnames)
grupos <- dados[,r]
structure(list
(grupos = grupos, centroide = centro, func.objetivo = f.opt))
#return(grupos)}
```

Apêndice B

Código do Algoritmo HBaseTabu2

Neste apêndice apresenta o código modificado em linguagem R de computação estatística da versão do Algoritmo HBaseTabu do Capítulo 2.

```
#########################################################
######## Uma versão do HBaseTabu2 ##############
#########################################################
H.BaseTabu2<- function(elementos,M,k,np,r,iter.max=15){
  elementos
  dados<-as.matrix(elementos)
  r<- ncol(dados)+1         # r é o número de colunas mais um #
  n<- nrow(dados)           # n é o número de linha das matriz #
  k                         # número de perturbações #
  np         # número funções objetivos das perturbações #
  iter.max                  # número máximo de iterações #
  elit<-0
  # matriz melit #
  ##### ncol2<-ncol(dados) + iter.max
  vmenores<- vector()
  melit <- matrix(nrow=n, ncol=iter.max)
  ####   v.menores <-  c(1:iter.max)
  # Condição para que a matriz seja validada #
  if(is.matrix(dados)){
  dados<-as.data.frame(dados)}
  ########  Passo 1  ##########
  # Partição Inicial #
  dados[,r] <- sample(1:M,n,replace=TRUE)
  ### Função Perturbação ###
  Perturbacao <- function(dados, k, r, M){
  # Escolhendo a posicao aleatoriamente #
  # Escolhendo aleatoriamente os clusters para as posicoes #
  vet<-as.vector(dados[,r])
  mat<-matrix(,ncol=length(dados[,r]),nrow=k)
  vet0<-vet
```

```
for(j in 1:k){
a<-sample(1:length(dados[,1]), j)
vet<-vet0
vet[a] <- sample(1:M, j, replace=TRUE)
mat[j,]<-as.factor(vet)}
final<- list(dados=dados, pert = t(mat),
com = cbind(dados,t(mat)))return(final) }
#### Função Objetivo para a partição inicial  #####
##  Função objetivo atual   ##
f.atual2 <- function(dados, M, r, n){
#Centroide
centro <<- matrix(0,nrow = M, ncol = (r-1))
for (i in 1:(r-1)){
centro[,i] <- tapply(dados[,i], dados[,r], mean)}
# Distancia entre cada elemento e sua centroide
distancias <- numeric()
for (elem1 in 1:n){
distancias[elem1] <- (dist(rbind(dados[elem1,1:(r-1)],
centro[(dados[elem1,r]), ]), method = "euclidean"))^2}
f.obj <<- sum(distancias)
structure(list( centroide = centro,  f.opt = f.obj ))}
# função objetivo da partição inicial #
jb <- f.atual2(dados,M,r,n)$f.opt
print(jb)
##  Funcao objetivo atual para as perturbações  ##
##  Calcula a funcao objetivo e a centroide ##
objetivo1<-c(1:np)
f.atual <- function(dados, M, r, n, np){
#Centroide
for(k in (r+1):(r+np)){
centro <- matrix(,nrow = M, ncol = (r-1))
for (i in 1:(r-1)){
centro[,i] <- tapply(dados[,i], dados[,k], mean)}
# Distancia entre cada elemento e sua centroide
distancias <- numeric()
for (elem1 in 1:n){
m<-rbind(dados[elem1,1:(r-1)],centro[(dados[elem1,k]), ])
distancias[elem1] <- (dist(m,method = "euclidean"))^2}
objetivo1[k-r]<<-sum(distancias)} }
######## Passo 2 #############
######## Fase da Diversificação ######
iter<-1
for(iter in 1:iter.max) {
perturbacao1<- Perturbacao(dados, k, r, M)
# Perturbacao de M grupos #
perturbacao1
```

```
# Matriz dos elementos, partição inicial e M grupos #
dados1<-perturbacao1$com
# funções objetivos de k perturbações #
funcaopertur<- f.atual(dados1, M, r, n, np)
# chamando o vetor dos valores das funções objetivos #
objetivo1
# o menor valor da função objetivo 1 ou jts #
jt<-min(objetivo1)
# a menor posição do valor das funções objetivos #
rt<-which.min(objetivo1)
# rt é somado com r #
novapert<-dados1[,(r+rt)]
# matriz dados com a partição inicial #
novapert
# substituindo a partição inicial por rt #
dados[,r]<-novapert
dados[,r]
dados
######## Passo 3 #########
## Primeira condição desse passo ##
if( jt < jb) { jb<-jt
#### Chamar H-Means ####
# Calcula-se H-Means para a matriz dados #
resultado<-
H.means2(dados[,-r], M, iter.max,initial.part = novapert)
resultado2<- resultado$func.objetivo
resultado2           # resultado da função objetivo #
particaoinicial<-resultado$grupos
agrupamento<-table(particaoinicial)
dados[,r]<-particaoinicial
jb<-resultado2 }
else {
####### Passo 4 #######
## FASE DA INTENSIFICAÇÃO ##
jc<-jt           # matriz de perturbações #
novapert<-dados1[,(r+rt)]      # rt #
dados[,r]<-novapert
dados[,r]
## soluções elite ##
elit<-elit+1
##v.menores <- as.vector(elit) ##
vmenores[elit]<-jc
melit[,elit] <- novapert}
iter<-iter + 1
list(print(jb), print(dados1),print(objetivo1),print(jt),
print(novapert),print(resultado2),
```

```
  print(particaoinicial),print(agrupamento),
  print(vmenores),print(melit))}
  ### INTENSIFICAÇÂO EM SOLUÇÕES-ELITE  #######
  int=0
  for (int in 1:3) {
  rc <- which.min(vmenores)
  print(rc)
  fc <- vmenores[rc]
  print(fc)
  ultimo <- melit[,rc]
  resultado3 <-
  H.means2(dados[,-r], M, iter.max,initial.part = ultimo)
  resultado4 <- resultado3$func.objetivo
  particaofinal <- resultado3$grupos
  list(print(resultado4), print(particaofinal))
  rmax <- which.max(vmenores)
  print(rmax)
  Z <- vmenores[rmax]
  print(Z)
  vmenores[rc] <- fc + Z
  print(vmenores)
  if(resultado4 < jb) {
  jb<-resultado4
  particaojb= particaofinal
  print(particaofinal)
  print(jb)} }
  ## resultado final - melhor solução encontrada ##
  list('melhor solucao',print(resultado4),
  print(particaofinal))}
##########################################################################
```